METHOD AND DEVICE FOR DATA COMMUNICATION

The present invention relates to a device and a method for data communication between a first host device or a further host device and at least one client device along a shared transmission path. A host device is a host computer (personal computer, workstation), for example, and a client device may be any arbitrary hardware application (e.g., FPGA or ASIC).

Figure 3 shows a communication system from the applicant of the present application, which is known under the trademark UMRBus. This communication system allows easy transmission of data and/or signals between a host device or host computer (300) and assigned hardware (clients), such as ASICs or FPGAs. Different software applications may communicate with their corresponding hardware applications via independent virtual channels, which use the same hardware interface between the host computer (300) and the client hardware (331, 332, 333). This communication system represents a simple standardized interface between the host computer (300) and at least one client (331, 332, 333). This client (331, 332, 333) may be a software application or a hardware application.

Individual host applications (301, 302, 303), which are indicated in Figure 1 with "HAPP" (host application), are connected via a host application interface (310) "HAPI" to a host application interface module (312) "HAPIM". The host application is a program which exchanges data with one or more client application(s) via the host application interface. The host application interface module (312) is a software library which provides multiple functions and commands for access on the communication system. A host application interface HAPI is an interface which host applications are placed on.

The host application interface module (312) is directly connected to a software interface module (314) "SIM", which typically corresponds to the device driver specific to the operating system. A host application interface module HAPIM provides the host application interface and transmits data from and to clients in cooperation with the software interface module.

The software interface module (314) is connected via a standardized interface connection (316), PCI or RS232, for example, to a hardware interface module (318) "HIM". The hardware interface module (318) represents the counterpart to the software interface module (314) in hardware. This hardware interface module (318) may, for example, have a PCI controller and an interface component between the PCI controller and a data bus. The software interface module SIM executes the data exchange between the host application interface module HAPIM and the hardware interface HIM.

An essential feature of the hardware interface module (318) is that it always provides a physical data bus (320) on the side away from the software. This data bus (320) connects the hardware interface module (318) to multiple client application interface modules (341, 342, 343) "CAPIM". In this case, the data bus (320) represents a ring connection between the client application interface modules (341, 342, 343) linked therein and the hardware interface module (318).

The hardware interface module HIM exchanges data with the software interface module SIM and makes the data bus available to the clients.

Each client application interface module (341, 342, 343) represents a type of node which is incorporated into the physical data bus (320). Each client application interface module (341, 342, 343) is connected via a client application interface (322) "CAPI" to a client application (351, 352, 353) "CAPP". Each client application interface module (341, 342, 343) has its own address and a type identity and provides the client application interface (322) for the client application (351, 352, 353).

In the device shown in Figure 3, all of the data communication is controlled by the host computer (300).

In this case, data exchange may only occur between one hardware interface module (318) and one client application interface module (341, 342, 343) at a time. Therefore, communication is only possible between the host computer (300) and the client applications (351, 352, 353), but not within the client applications (351, 352, 353), implemented in the example shown as hardware. In addition, this known device for data communication may not be used in systems having multiple host computers.

The object of the present invention is to provide a device for data communication between a first host device or a further host device and at least one client device along a shared transmission path, as well as a corresponding method of data communication.

The object relating to the device is achieved by the features specified in Claim 1.

Using this device according to the present invention, data communication is made possible on a shared transmission path even if multiple host devices are provided. For example, this device may be used within image processing systems based on FPGA, in order to control the image processing implemented in FPGA, even if the image processing system has its own integrated processor, i.e., its own host device, which may, using the data bus, wish to change parameters in parallel to the higher order host device. In this case, the bus control module in the device according to the present invention assumes the task of controlling which of the host devices may use the transmission path in a specific period of time or for a specific quantity of data.

The first and/or the further host device, particularly the host application, preferably has a processor.

The transmission path is preferably implemented in this case as a data bus. Advantageous refinements of this device are specified in Claims 3 through 5.

The object relating to the method is achieved by the method steps specified in Claim 6.

An essential feature of this method according to the present invention is the provision and transmission of arbitration information on the transmission path along the open communication connection, the arbitration information ensuring that only one host device receives access to the transmission path at any time. Preferred embodiments and refinements of the method according to the present invention are specified in Claims 7 through 10.

The present invention is described in greater detail in the following on the basis of an example with reference to the drawing.

Figure 1 shows the schematic illustration of a device for data communication according to the present invention;

Figure 2 shows a flow chart of data communication according to the method according to the present invention, and

Figure 3 shows the schematic construction of a device for data communication according to the related art.

Figure 1 shows a device for data communication according to the present invention having two host devices 1, 2. The first host device 1 has a personal computer (PC) 11, for example, and the second host device 2 has a processor integrated in a hardware application (embedded processor) 21, for example. A hardware application of this type having an integrated processor may be an ASIC or an FPGA having an integrated processor, for example. FPGAs of this type are used in image processing systems, for example. Thus, for example, parameters for a frame grabber are transmitted from the first host computer to the FPGA hardware, which has the integrated processor.

The first host computer 1 and the second host computer 2 are each provided with a master application 14, 24 "MAPP", which is linked in a data bus 9 via a master application interface module "MAPIM".

The master application 14, 24 is an application implemented in hardware in this example, which is connected via a master application interface 12, 22 "MAPI" to the master application interface module 10, 20 and via this to the data bus 9.

Furthermore, five client devices 3, 4, 5, 6, 7 are provided, for example, which each have a client application "CAPP" 34, 44, 54, 64, 74. Each client application is connected via a client application interface "CAPI" 32, 42, 52, 62, 72 to a client application interface module "CAPIM" 30, 40, 50, 60, 70, which are each connected into the data bus 9.

In this case, a client application is an application implemented in hardware which, via the client application interface 32, 42, 52, 62, 72, exchanges data via the data bus 9 with one of the host computers 1, 2. This data exchange is controlled by the corresponding master application 14, 24.

The client application interface modules 30, 40, 50, 60, 70 each form the interface between the data bus 9 and the particular client application interface 32, 42, 52, 62, 72. The particular clients 3, 4, 5, 6, 7 connected via these client application interfaces 32, 42, 52, 62, 72 to the data bus 9 and/or the client applications 34, 44, 54, 64, 74 running thereon are only provided with the data and services which relate to the particular client application 34, 44, 54, 64, 74.

Furthermore, a bus control module "BCM" 8, which administrates the access of the individual master application interface modules 10, 20 to the data bus 9 and monitors the data bus interrupts, is linked in the data bus 9.

In this device according to the present invention, the hardware interface module HIM is not, as in the related art shown in Figure 3, linked directly in the data bus, but provides the data on the data bus 9 via the master application interface module 10, 20. The hardware interface module HIM may be formed in this case by the particular master application MAPP 14, 24.

In the following, the access administration of a host device to a client device is described on the basis of Figure 2.

At the beginning of operation, the bus control module BCM 8 transmits arbitration information which is continuously available on the data bus 9. This arbitration information is, for example, implemented as an arbitration frame which contains a frame header and a data word.

If a host computer wishes to access the data bus and transmit data to or exchange data with a client application via the bus, it indicates this access wish to the master application interface module MAPIM assigned to it in step 101. The master application interface module MAPIM then accepts the arbitration frame relayed on data bus 9 in step 102 and reads out the arbitration data therefrom in step 103. An activity bit is included in the arbitration data which specifies whether the data bus 9 is currently used by another application. If this activity bit is not set, this indicates that the data bus 9 is free.

In step 104, the master application interface module MAPIM now checks whether the activity bit is set or not. If the activity bit is set, i.e., the data bus is not free, the MAPIM relays the arbitration frame further to the data bus in step 105 and accepts it again after a pause 106 in order to repeat steps 102 to 104.

If, however, it was determined in step 104 that the activity bit is not set, i.e., the data bus is free, the master application interface module MAPIM sets the activity bit in turn in step 107 and thus indicates that it wishes to access the data bus 9. The arbitration frame is then transferred to the data bus 9 again in step 108 and runs further there until it reaches the bus control module BCM 8. This module accepts the arbitration block in step 109, reads out the arbitration data in step 110, and reserves the data bus 9 for the querying host computer in step 111.

A master application interface module MAPIM therefore may only use the data bus when it receives the arbitration frame and the activity bit has not already been set therein by another master application interface module. The arbitration frame is received by the bus control module BCM and is not given back to the data bus if the activity bit is set. The master application interface module MAPIM which has set the activity bit may now transmit data via the data bus 9.

After termination of this releasing data communication via the data bus 9 or after expiration of the predetermined time interval or the predetermined data volume for data transmission, the master application interface module MAPIM transmits the arbitration frame again, but now with the activity bit deactivated, which indicates that the data bus 9 is now free again. In this way, it is ensured that only one master application interface module MAPIM is active on the data bus 9 at a time and overlaps of data transmissions on the data bus do not occur. The bus control module BCM 8 monitors the data bus 9 in that it counts the pulses between the passage of two arbitration frames using a counter. If the count exceeds the predetermined limiting value of the number of pulses per access, it is concluded therefrom that no arbitration frame was received. This may occur, for example, due to an interruption of the data bus 9 or a reset of the entire device. In this case, the bus control module 8 retransmits an arbitration frame having a deactivated activity bit and thus automatically releases the blocked data bus 9 again.

If an interrupt is displayed on the data bus, the bus control module BCM 8 takes over the data bus 9 and transmits an interrupt frame. If the interrupt frame was successfully transmitted, the bus control module BCM 8 transmits an "interrupt information frame". Using this "interrupt information frame", the master application interface module MAPIM is informed about the interrupt status of the individual client application interface modules CAPIM and may relay this information to the particular master application MAPP for further processing.

In this way, it is ensured that all host computers are informed about interrupts.

In this way, a device according to the present invention for data communication, using a data communication method running thereon, with a multimaster-capable data bus 9 is provided.

The present invention is not restricted to the exemplary embodiment above, which is merely used for general explanation of the core idea of the present invention. Rather, within the scope of protection, the device according to the present invention may also assume other embodiments than those described above. In particular, the device may have features which represent a combination of the particular individual features of the claims.

Reference numbers in the claims, the description, and the drawing are merely used for better understanding of the present invention and are not to restrict the scope of protection.

LIST OF REFERENCE NUMBERS

1	host computer
2	host device
3	client device
4	client device
5	client device
6	client device
7	client device
8	bus control module BCM
9	data bus
10	master application interface module MAPIM
12	master application interface module MAPI
14	master application MAPP
20	master application interface module MAPIM
22	master application interface module MAPI
24	master application MAPP
30	client application interface module CAPIM
32	client application interface module CAPI
34	client application CAPP
40	client application interface module CAPIM
42	client application interface module CAPI
44	client application CAPP
50	client application interface module CAPIM
52	client application interface module CAPI
54	client application CAPP
60	client application interface module CAPIM
62	client application interface module CAPI
64	client application CAPP
70	client application interface module CAPIM
72	client application interface module CAPI
74	client application CAPP

101, 102,103,104,105, 106,107,108,109, 110, 111:
method steps

300 host device

301	host application HAPP
302	host application HAPP
303	host application HAPP
310	host application interface HAPI
312	host application interface module HAPIM
314	software interface module SIM
316	interface connection
318	host interface module
320	data bus
322	client application interface CAPI
331	client device
332	client device
333	client device
341	client application interface module CAPIM
342	client application interface module CAPIM
343	client application interface module CAPIM
351	client application CAPP
352	client application CAPP
353	client application CAPP